

Modified NSGA-II for a Bi-Objective Job Sequencing Problem

Susmita Bandyopadhyay

Department of Production Engineering, Jadavpur University, Kolkata, India
Email: bandyopadhyaysusmita2009@gmail.com

Received July 8, 2012; revised August 9, 2012; accepted October 12, 2012

ABSTRACT

This paper proposes a better modified version of a well-known Multi-Objective Evolutionary Algorithm (MOEA) known as Non-dominated Sorting Genetic Algorithm-II (NSGA-II). The proposed algorithm contains a new mutation algorithm and has been applied on a bi-objective job sequencing problem. The objectives are the minimization of total weighted tardiness and the minimization of the deterioration cost. The results of the proposed algorithm have been compared with those of original NSGA-II. The comparison of the results shows that the modified NSGA-II performs better than the original NSGA-II.

Keywords: Job Sequencing; Multi-Objective Evolutionary Algorithm (MOEA); NSGA-II (Non-Dominated Sorting Genetic Algorithm-II); Tardiness; Deterioration Cost

1. Introduction

Job sequencing is a problem of arranging jobs in a sequence so as to minimize makespan, tardiness, completion time, waiting time, idle time and so on. The problem of job sequencing is traditional and thus quite a significant number of research studies are observed in the existing literature. In job sequencing, finding the right sequence to satisfy the above constraints for a multiple job and multiple machines is still a challenging research study. Numerous research efforts are observed towards this direction.

A variety of objectives for job sequencing problems have been investigated by the researchers in the relevant field of study and sometimes, similar set of objectives are observed in more than one research study, solving such problems with various methods. Both deterministic and non-deterministic methods have been applied to solve these types of problems. The deterministic methods include the mathematical methods like Linear Programming, Dynamic Programming, Integer Programming and so on. The non-deterministic methods mainly include various nature based algorithms like Genetic Algorithm, particle Swarm Optimization, Ant Colony Optimization, Simulated Annealing, Frog Leaping Algorithm, Bee Colony Algorithms and so on. The use of simulation techniques is also observed to solve job sequencing problems.

In this paper, a bi-objective job sequencing problem

has been formulated and a modified version of NSGA-II (Non-dominated Sorting Genetic Algorithm-II) have been proposed to solve the formulated problem. The problem has also been solved by the original NSGA-II and the results of the two algorithms have been compared.

The remaining sections of this paper are organized in the following way. Section 2 reviews the existing literature; Section 3 formulates the problem; Section 4 describes the proposed modified NSGA-II; Section 5 shows results and discussion; Section 6 concludes this paper.

2. Literature Review

Job sequencing is a traditional area of research. Thus a significant number of research studies are observed in the existing literature.

The minimization of makespan (Xia and Wu [1], Xing *et al.* [2], Miao *et al.* [3]) is observed to be very common in the existing literature. Xia and Wu [1], Xing *et al.* [2], Li *et al.* [4], Zhang *et al.* [5], Li *et al.* [6], Gao *et al.* [7] considered total workload of machines and maximum workload as objectives. In addition to these objectives, Xia and Wu [1] and Gao *et al.* [7] considered minimization of makespan and Li *et al.* [4] considered maximum completion time. Rahimi-Vahed and Mirzaei [8] and Rahimi-Vahed *et al.* [9] worked with three objectives—minimization of total utility work, minimization of total production rate variation and minimization of total setup

cost.

Tay and Ho [10] considered three objectives—minimization of makespan, mean tardiness and mean flow time. The minimization of makespan, total tardiness and total idle time was considered by Sha and Lin [11] and Mattfeld and Bierwirth [12] minimized weighted mean flow time, weighted mean tardiness, maximum tardiness and weighted number of tardy jobs. Among the other significant research studies considering tardiness, makespan and flow time as the minimization objectives include the research studies of Varadharajan and Rajendran [13], Yagmahan and Yenisey [14], Behnamian *et al.* [15], Chiang *et al.* [16].

In the work of Lian [17], the objectives considered were—minimization of runtime of every machine, earliness time and process time of jobs. Tavakkoli-Moghaddam *et al.* [18] considered the minimization of weighted mean completion time and weighted mean tardiness. Mazdeh *et al.* [19] dealt with the minimization of total job tardiness and total machine deterioration cost.

Numerous solution methodologies have been considered for solving the job sequencing problems in the existing literature. The methodologies include Linear Programming (Mazdeh *et al.* [19]), dynamic programming (Lewis and Slotnick [20]), Integer Programming (Baker and Keller [21]), Genetic Algorithm (Lin and Jia Zhen [22], Mattfeld and Bierwirth [12], Gao *et al.* [7]), Frog Leaping algorithm (Xing *et al.* [2]), Rahimi Vahed and Mirzaei [8], Li *et al.* [4]), Particle Swarm Optimization (PSO) (Xia and Wu [1], Zhang *et al.* [5], Sha and Lin [11]), Ant Colony Optimization (ACO) (Yagmahan and Yenisey [14], Yagmahan and Yenisey [23], Huang [24]), Simulated Annealing (SA) Lian [17], hybrid algorithms (Zhang and Wu [25], Tavakkoli-Moghaddam *et al.* [18], Wang *et al.* [26]). Besides, Moradi *et al.* [27] solved bi-objective job sequencing problem by NSGA-II (Non-dominated Sorting Genetic Algorithm-II) [28] and NRGGA (Non Ranking Genetic Algorithm) algorithms.

In this paper, the original NSGA-II has been modified by embedding a new mutation algorithm in the original NSGA-II. The existing literature shows a variety of improvements to NSGA-II, such as controlled elitism [29, 30], scalarizing fitness function [31], application of sequential quadratic programming [32], application of nearest neighbor approach [33], various sorting methods [34,35], various distribution methods [36], various diversity preservation methods [37] and so on.

A number of crossover and mutation techniques are also observed in the existing literature. Some of the common crossover techniques are One-Point Crossover, Uniform Crossover, Partially Mapped Crossover, Order Crossover, Cycle Crossover, Simulated Binary Crossover, Position Based Crossover, Parent Centric Recombination

and so on. An extensive study of the literature indicates that no particular crossover or mutation technique is universally effective to all types of problems. Both the crossover and mutation algorithms depend on the type of the problem under study and the variables used in a problem.

The bi-objective job sequencing problem considered in this paper has been solved by two algorithms—NSGA-II and a modified version of NSGA-II as proposed in this paper.

3. Problem Formulation

Before formulating the problem, the assumptions and the notations are listed below.

3.1. Assumptions

Following are the assumptions made for formulating the problem:

- 1) Each job is processed on more than one machine;
- 2) Each job has a processing time and the processing times of the jobs are not same;
- 3) Processing time of a job varies with different machine;
- 4) Each machine can process only one job at a time;
- 5) Deterioration depends on both job and machine;
- 6) For each job, for each machine, there is a separate deterioration cost.

3.2. Notations

3.2.1. Decision Variables

x_{ijm} : 1 if job j follows job i in sequence on machine m and 0 otherwise;

y_{jm} : 1 if job j is assigned to machine m and 0 otherwise.

3.2.2. Parameters

W_i : Weight related to i -th job;

T_i : Tardiness of the i -th job;

c_i : Completion time of the i -th job;

d_i : Due date of i -th job;

R_{jm} : Machine deterioration cost for job j on machine m ;

J : Total number of jobs;

M : Total number of machines;

j : Subscript for jobs;

m : Subscript for machine;

S_{jm} : Staring time of job j on machine m ;

P_{jm} : Processing time of job j on machine m .

3.3. Formulated Problem

$$\text{Min}Z_1 = \sum_{i=1}^J W_i T_i \quad (1)$$

$$\text{Min}Z_2 = \sum_{m=1}^M \sum_{j=1}^J R_{jm} y_{jm} \quad (2)$$

Subject to the constraints:

$$\sum_{\substack{i=1, i \neq j \\ j=1}}^J x_{ijm} = 1 \quad (3)$$

$$\sum_{j=1, i \neq j}^J x_{ijm} \leq y_{im} \quad (4)$$

$$\sum_{m=1}^M y_{im} = 1 \quad (5)$$

$$c_j \geq S_{jm} + P_{jm} \quad (6)$$

$$T_i \geq c_i - d_i \quad (7)$$

$$x_{ijm} + x_{jim} = 1 \quad (8)$$

$$c_i, S_{jm}, T_i \geq 0 \quad (9)$$

Objectives (1) and (2) minimize the total weighted tardiness of all jobs and the total deterioration cost of all the jobs respectively. Constraint (3) indicates that only one job precedes each job. Constraint (4) conveys that if job j follows job i then both job i and job j belong to machine m , assuming that only one job follows a job and only one job precedes a job. Constraint (5) states that each job is assigned to exactly one machine. Constraint (6) means that the completion time of job j is greater than or equal to the sum of the starting time of job j on machine m and the processing time of job j on machine m . Constraint (7) defines the tardiness. Tardiness of a job is the positive lateness of the job which is calculated by subtracting the due date from the completion of job j . Constraint (8) ensures that either job i will follow job j or job j will follow job i . Constraint (9) ensures that c_i , S_{jm} and T_i must be positive quantities.

4. Modified NSGA-II

In this paper, a modification of a popular Multi-objective Evolutionary Algorithm (MOEA) known as NSGA-II (Non-Dominated Sorting Genetic Algorithm-II) [28] is proposed. NSGA-II has been selected because of its population based nature and non-domination sort which assigns rank and crowding distance to each individual (chromosome) in the population. Besides, NSGA-II is the most widely applied MOEA as observed in the existing literature. The algorithm for the modified NSGA-II is shown in **Figure 1**. The algorithm continues to execute till the maximum number of generations. The main components of the algorithm are summarized below.

4.1. Initialization

The genotype of the chromosome is shown in **Figure 2**.

The job sequence is generated through random number generation. For each gene of the job sequence, a random number is generated and based on this random number, a job number is generated. A flag is set if a particular gene is assigned a job number. This flag prevents redundant job assignment to the genes for a particular chromosome. Next the machines are assigned to the jobs of the generated job sequence. The algorithm for the initialization is shown in **Figure 3**.

4.2. Non-Dominated Sorting

Before selection is performed, every individual in the population is assigned a rank based on non-domination: All non-dominated individuals are classified into one

```

INITIALIZE VARIABLES
EVALUATE OBJECTIVES
PERFORM NON-DOMINATION SORT OVER THE POPULATION TO ASSIGN
RANK AND CROWDING DISTANCE VALUE TO EACH INDIVIDUAL

FOR EACH GENERATION
IF PROBABILITY < Pc THEN
PERFORM TOURNAMENT SELECTION TO GENERATE MATING POOL
//PERFORM CROSSOVER OVER THE ENTIRE POPULATION
PERFORM ORDER CROSSOVER TO GENERATE OFFSPRING
COMBINE OFFSPRING POPULATION WITH ORIGINAL POPULATION TO
FORM INTERMEDIATE POPULATION
PERFORM NON-DOMINATION SORT OVER THE ENTIRE POPULATION
TAKE THE BEST CHROMOSOMES BASED ON RANK AND CROWDING
DISTANCE TO FILL THE ORIGINAL SIZE OF POPULATION
ELSE
PERFORM MUTATION OVER THE ENTIRE POPULATION
PERFORM NON-DOMINATION SORT
END IF
END FOR

```

Figure 1. Modified NSGA-II algorithm (Pc: Crossover Probability).

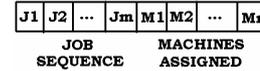


Figure 2. Chromosome Representation.

```

FOR EACH MEMBER OF POPULATION

FOR EACH j ∈ J
SET flag[j] ← 0
END FOR

FOR EACH j ∈ J
GENERATE RANDOM NUMBER r
GENERATE JOB NUMBER n BASED ON r
IF flag[n] != 1 THEN
ASSIGN Job[j] ← n
SET flag[n] = 1
END IF

```

Figure 3. Algorithm for initialization (J: Number of jobs in a job sequence).

category (with a dummy fitness value, which is proportional to the population size, to provide an equal productive potential for these individuals). The crowding distance is also calculated (see Equation (10) below) to keep a diverse front by making sure that each member stays a crowding distance apart. This keeps the population diverse and helps the algorithm to explore the fitness landscape.

$$d_I = \sum_{i=1}^B \frac{f_i(x) - F}{f_i^{\max} - f_i^{\min}} \quad (10)$$

where d_I : crowding distance for individual (chromosome) I ; f_i^{\max} and f_i^{\min} are the maximum and minimum objective values of the i -th objective respectively.

Non-dominated sorting is performed both after crossover and after mutation in the modified NSGA-II proposed in this paper. In case of crossover operation to be performed, the individuals are selected to form a mating pool. The selection is done through Tournament selection method. The mating pool contains the individuals on which the crossover is performed.

4.3. Crossover

Based on the structure of chromosome, order crossover has been applied in this paper. The algorithm for order crossover as applied in this paper is shown in **Figure 4**.

In order crossover, as applied here, two crossover sites are generated randomly and the genes in between these crossover sites are copied from one of the parents to one of the children and the same genes in the other parent, are nullified. The remaining genes in the second parent are copied to the same child, in the same order as in the second parent. In this way, one of the children is generated. The other child is generated in the same way, by exchanging parents. **Figure 4** shows the generation of any one child from the parents.

4.4. Mutation

The mutation algorithm is shown in **Figure 5**.

The mutation is performed on the entire population. For each variable, the alleles (values of genes) representing a variable are summed up. Then the entire popu-

```

SELECT TWO PARENT P1 AND P2
RANDOMLY GENERATE THE FIRST CROSSOVER SITE S1
RANDOMLY GENERATE THE SECOND CROSSOVER SITE S2
COPY THE ALLELES (VALUE OF GENES) BETWEEN S1 AND S2
FROM P1 TO C1 (CHILD 1) ON THE SAME GENE POSITIONS
IDENTIFY THE POSITIONS OF THESE GENES IN P2 AND ASSIGN
NULL TO THESE GENE POSITIONS IN P2
COPY THE OTHER GENES OF P2 TO THE EMPTY GENE
POSITIONS OF C1 IN THE SAME ORDER AS IN P2

```

Figure 4. Order crossover to generate an offspring.

```

FOR EACH VARIABLE  $v$ 
FIND sum OF ALL GENES FOR  $v$ 
DIVIDE THE POPULATION OF INDIVIDUALS INTO  $G$ 
DIVISIONS, WHERE  $G$  IS THE NUMBER OF JOBS
COUNT THE NUMBER OF INDIVIDUALS  $cnt_i$  IN EACH DIVISION
FIND THE MAXIMUM  $max$  OF ALL  $cnt_i$ 
CALCULATE  $num = N / I$  WHERE  $I$  IS THE TOTAL NUMBER OF
DIVISIONS
FOR EACH  $cnt_i$ 
IF  $cnt_i$  EQUALS  $max$  THEN
  CALCULATE  $diff = max - min$ 
  FIND AN INDIVIDUAL  $d'$  BELONGING TO THE DIVISION
  OF  $cnt_i$ 
  FOR  $j = 1$  TO  $diff$ 
    FIND A DIVISION  $g_j \in G$  WHICH IS DIFFERENT FROM
    THE DIVISION OF  $cnt_i$ 
    MUTATE  $d'$  TO THE RANGE OF DIVISION  $g_j$ 
  END FOR
END IF
END FOR

```

Figure 5. Mutation algorithm (N : population size).

lation is divided into a number of divisions. The number of divisions is equal to the number of jobs in a job sequence. Then the number of individuals (chromosomes) in each division is counted and the maximum *max* of these counts is found out. The individuals in the division containing the maximum number of individuals are mutated to the divisions containing less number of individuals. The number of individuals mutated is the difference between *max* and the ideal number of individuals that should have been present in each division if the population would be divided equally into these divisions.

5. Results and Discussion

The experiments have been conducted in a PC with 2.8 GHz processor and 1 GB memory. Matlab has been used to program both the original NSGA-II and the modified NSGA-II algorithms. For each of these two algorithms, the program has been run for 10 generations starting from generation 10 to generation 100. For each of these generations, nine crossover probabilities starting from 0.1 up to 0.9 have been applied. For each of these probabilities and generations, the program has been run 10 times and the best results have been taken. Thus the total number of experiments conducted is: 10 (number of gen-

erations) \times 9 (number of probabilities) \times 10 (number of runs for each of these values) = 900, for each of the two algorithms.

For experimentation, a total of 6 jobs and 4 machines have been assumed. Separate deterioration cost for each job-machine pair has been considered and tardiness has been calculated based on the processing times and due dates. The population size has been taken to be 100. Based on the number of experimentations conducted, a considerable volume of results have been obtained, although it has been observed that the best results have been obtained for probabilities 0.5 to 0.9 and the results have been significant from generations 50 to generation 80. Thus the Pareto optimal solutions, in particular, are shown here for generations 50 to generation 80, for probabilities 0.5 to 0.9.

Figures 6-9 show Pareto optimal solutions for generations 50, 60, 70 and 80 respectively. The horizontal and vertical axes represent objective 1 and objective 2 respectively. **Figure 6** shows nearly similar results for both the original NSGA-II and modified NSGA-II. **Figure 7** shows better results for the modified NSGA-II over the original NSGA-II for all probabilities, especially for the second objective. **Figure 8** shows better results on

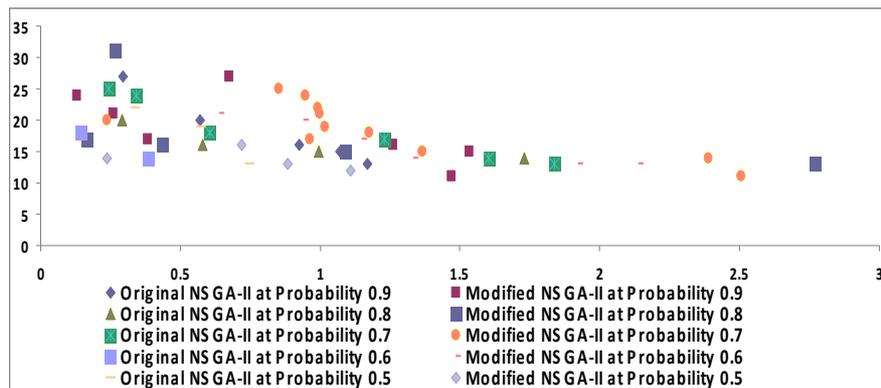


Figure 6. Pareto optimal solutions for generation 50.

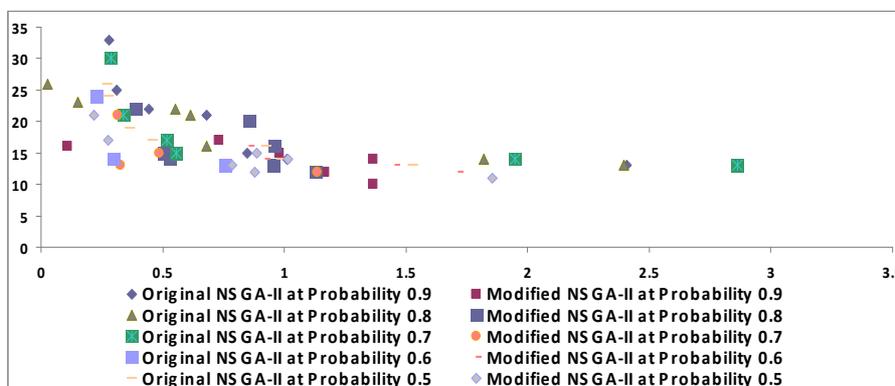


Figure 7. Pareto optimal solutions for generation 60.

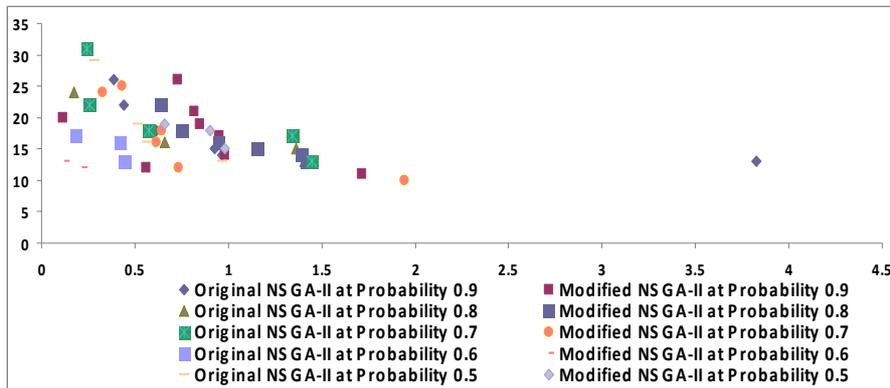


Figure 8. Pareto optimal solutions for generation 70.

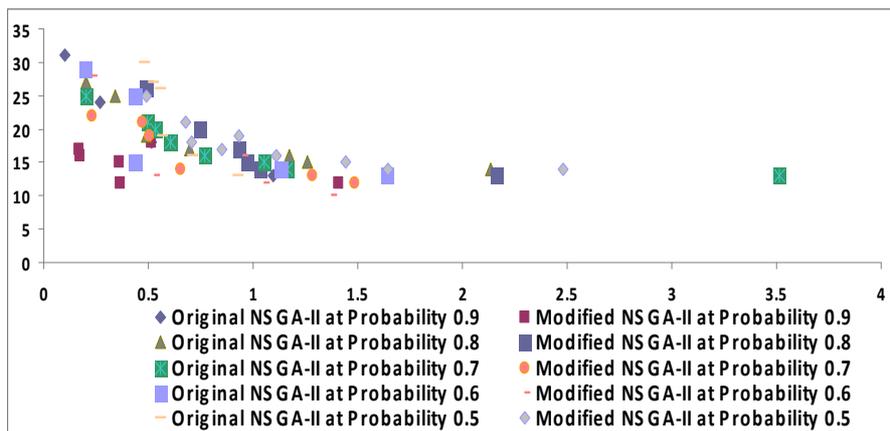


Figure 9. Pareto optimal solutions for generation 80.

the average, for the original NSGA-II, whereas **Figure 9** shows better result for the modified NSGA-II. Thus in aggregate, **Figures 6-9** show better result for the modified NSGA-II.

Tables 1-4 also show minimum values of both objective 1 and objective 2 for both the algorithms. These values have been calculated over the entire population of solutions. **Tables 1** and **2** show minimum values of objective 1 for the original NSGA-II and the modified NSGA-II respectively, whereas, **Tables 3** and **4** show minimum values of objective 2. The comparisons of the values between **Tables 1** and **2** as well as between **Tables 3** and **4** show better results for the modified NSGA-II.

Tables 5 and **6** compare the execution times of the two algorithms and it is clearly observed that the values in **Table 5** for the modified NSGA-II are lower (better) than the values in **Table 6** (values for original NSGA-II).

The figures and tables in the following parts show the performance of the two algorithms and the modified NSGA-II performs better than the original NSGA-II. The differences in terms of performance between the original NSGA-II and the modified NSGA-II, as observed from

the experimental results, are summarized in **Table 7**. **Table 7** shows that the modified NSGA-II performs better than the original NSGA-II in terms of the Pareto Optimal solutions, the minimum values of the objectives and the execution time, which in turn, indicate better search capability of the modified NSGA-II over the original NSGA-II. Thus the introduction of the mutation algorithm and the application of the mutation algorithm over the entire population are found to be effective in order to improve the overall performance of the NSGA-II algorithm, as a whole.

However the purpose of the experimentation was to observe the variety of results which have been shown by both the algorithms and this indicates that both of these algorithms can be used to solve such a job sequencing problem. However two subsets of the Pareto optimal solutions are shown in **Tables 8** and **9** for the modified NSGA-II and the original NSGA-II respectively.

6. Conclusion

This paper formulates a multi-objective job sequencing problem consisting of two objectives which are minimization of total weighted tardiness and minimization of

Table 1. Minimum values of objective 1 for original NSGA-II.

Gen.	Probabilities								
	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
10	0.4817	0.8351	0.6812	0.6903	0.74982	0.6723	0.423	0.5707	0.5933
20	0.2313	0.6285	0.6396	0.2948	0.1374	0.8502	0.5256	0.1791	0.6947
30	0.5409	0.5028	0.1905	0.3116	0.4468	0.5354	0.4586	0.1246	0.2368
40	0.2503	0.2008	0.3585	0.313	0.1467	0.2541	0.2535	0.3241	0.6887
50	0.2959	0.2929	0.2447	0.1457	0.3376	0.309	0.4575	0.2092	0.3894
60	0.2814	0.5517	0.2886	0.2287	0.2729	0.3051	0.3272	0.3362	0.5851
70	0.3851	0.1713	0.2423	0.1857	0.2841	0.1025	0.2528	0.2217	0.4531
80	0.202	0.621	0.1997	0.1955	0.4806	0.784	0.231	0.3303	0.413
90	0.3695	0.1345	0.657	0.3804	0.3028	0.914	0.2446	0.2523	0.1797
100	0.931	0.1029	0.841	0.1662	0.2416	0.1666	0.1801	0.1576	0.2279

Table 2. Minimum values of objective 1 for modified NSGA-II.

Gen.	Probabilities								
	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
10	0.3863	0.557	0.2676	0.6825	0.1983	1.0648	0.4179	0.4756	0.3413
20	0.5999	0.1589	0.5041	0.3715	0.124	0.407	0.6087	0.1916	0.7634
30	0.5186	0.4223	0.257	0.2728	0.1328	0.1989	0.6023	0.1047	0.1589
40	0.2279	0.1803	0.2271	0.1495	0.1811	0.2035	0.1867	0.4835	0.1178
50	0.1301	0.1638	0.2365	0.1411	0.2361	0.1551	0.1505	0.2736	0.199
60	0.1105	0.3901	0.316	0.2055	0.2166	0.9347	0.4341	0.3087	0.1996
70	0.1156	0.1675	0.3271	0.1258	0.2598	0.9626	0.119	0.1872	0.1931
80	0.1663	0.4883	0.1096	0.1302	0.4888	0.6562	0.1423	0.3168	0.1671
90	0.1542	0.1163	0.519	0.3166	0.2152	0.7633	0.1436	0.1839	0.1504
100	0.1005	0.1026	0.5536	0.1629	0.1095	0.1559	0.172	0.1073	0.1215

Table 3. Minimum values of objective 2 for original NSGA-II.

Gen.	Probabilities								
	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
10	13	15	14	15	14	13	16	14	14
20	13	13	14	14	14	15	13	14	14
30	14	13	14	16	14	15	14	14	15
40	14	13	13	14	13	14	14	18	14
50	13	14	13	14	13	13	13	14	13
60	13	13	13	13	13	13	13	13	14
70	14	15	13	14	16	13	15	16	14
80	13	14	19	13	13	14	14	13	14
90	13	14	13	14	13	14	14	14	14
100	13	13	14	14	18	14	14	13	14

Table 4. Minimum values of objective 2 for modified NSGA-II.

Gen.	Probabilities								
	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
10	12	15	13	11	15	15	13	14	16
20	13	11	14	13	14	14	15	14	13
30	11	13	12	14	12	14	13	14	13
40	14	11	14	13	12	12	15	17	15
50	11	13	11	13	12	15	13	13	13
60	10	13	13	13	11	13	16	14	12
70	11	13	10	12	15	15	14	15	13
80	12	13	12	10	14	16	12	13	14
90	10	13	12	13	12	12	13	13	14
100	11	13	14	12	15	16	12	10	14

Table 5. Execution times for the modified NSGA-II.

Prob.	Generation									
	10	20	30	40	50	60	70	80	90	100
0.9	3.344	6.485	9.781	12.547	16.235	19.359	23.063	25.765	29.469	33.094
0.8	2.984	6.328	9.562	12.797	14.657	19.547	22.484	24.516	28.125	31.703
0.7	3.187	6.078	9.109	12.219	14.812	17.234	21.719	24.437	28.343	31.062
0.6	3.359	5.5	9.187	11.265	13.485	17.062	19.875	23.594	26.25	28.39
0.5	2.562	5.578	8.312	11.063	13.375	17.187	19.094	22.265	24.656	27.25
0.4	2.454	5.469	7.218	9.953	13.25	14.703	17.343	18.062	22.781	25.219
0.3	2.828	5.235	7.985	9.125	10.75	13.203	16.25	20.516	20.703	23.407
0.2	2.25	4.484	6.875	9.219	10.937	12.5	16.156	16.687	18.781	21.344
0.1	2.14	4.828	6.125	8.5	10.641	11.672	13.969	15.625	17.391	20.172

Prob.: Probability.

Table 6. Execution times for the original NSGA-II.

Prob.	Generation									
	10	20	30	40	50	60	70	80	90	100
0.9	4.109	8.046	12.203	15.844	19.844	23.922	28.032	32.219	35.953	40.719
0.8	4.266	7.89	11.39	15.843	20.625	24.329	27.828	31.25	36.312	40.703
0.7	4.016	8.235	11.765	16.187	19.593	23.843	27.735	33.656	37.344	41.797
0.6	4.235	8.078	12.031	16.281	21.047	24.047	29.531	31.406	36.593	41.219
0.5	4.078	7.953	12.187	15.922	19.625	23.688	28.14	32.437	36.937	41.11
0.4	4.11	7.844	12.203	16.234	20.281	23.718	29.313	34.422	37.562	39.906
0.3	4.125	7.953	12.016	16.984	21.234	23.375	27.828	33.157	37.953	43.328
0.2	4	8.485	12.234	15.656	20.031	24.531	28.719	31.141	35.297	39.14
0.1	4.015	7.937	12.313	16.875	19.906	24.359	28.735	31.219	37.891	44.156

Prob.: Probability.

Table 7. Summary of aggregate differences in performance.

	Original NSGA-II	Modified NSGA-II
Pareto Optimal Solutions	Worse	Better
Minimum Values of Objective 1	Worse	Better
Minimum Values of Objective 2	Worse	Better
Execution Time	Worse	Better

Table 8. Pareto optimal solutions (job sequence) for generation 60 and probability 0.7 for modified NSGA-II.

2	6	3	1	4	5
4	2	6	5	1	3
6	1	3	5	2	4
3	2	6	1	4	5
2	3	1	6	4	5
6	2	1	3	4	5
6	2	1	3	4	5
2	1	6	5	4	3
3	1	6	4	5	2
6	2	1	3	5	4
2	1	3	6	4	5
4	3	6	2	5	1
2	1	6	3	4	5
3	6	2	1	4	5
2	1	6	4	5	3
1	6	3	2	4	5
4	1	6	3	2	5
2	3	1	6	5	4
5	1	2	3	4	6
4	1	6	3	2	5
2	1	3	6	5	4
3	1	6	2	4	5
1	6	2	4	5	3
4	5	6	2	1	3
3	1	6	4	2	5
2	1	6	3	5	4
2	1	6	3	5	4
2	4	6	1	5	3
6	2	1	3	4	5
1	3	6	2	5	4
4	1	6	2	3	5
1	6	2	3	4	5
5	1	6	2	4	3
5	1	6	2	3	4
1	3	6	5	2	4
3	2	6	5	1	4
2	1	6	4	5	3
1	2	6	3	5	4
4	6	1	2	5	3
1	6	2	4	5	3

Table 9. Pareto optimal solutions (job sequence) for generation 60 and probability 0.7 for original NSGA-II.

1	6	3	5	2	4
1	6	3	5	2	4
3	4	1	2	6	5
1	6	3	5	4	2
6	1	3	5	2	4
6	1	3	5	4	2
3	1	6	5	2	4
6	1	3	5	4	2
1	6	3	5	2	4
1	6	3	5	4	2
3	4	6	2	1	5
1	6	3	2	5	4
1	6	3	5	4	2
3	4	6	2	1	5
1	6	3	5	2	4
6	1	3	2	5	4
2	3	6	1	5	4
3	6	1	5	2	4
6	1	3	5	4	2
6	3	1	5	2	4
1	6	3	5	4	2
1	6	3	2	5	4
3	4	1	2	6	5
3	4	6	5	1	2
1	6	3	5	2	4
1	6	3	2	5	4
6	1	3	4	2	5
1	6	3	2	5	4
1	6	3	5	4	2
3	4	6	2	1	5
1	6	3	5	4	2
1	6	3	5	4	2
6	4	3	1	2	5
4	3	6	2	1	5

the deterioration cost. A popular Multi-Objective Evolutionary Algorithm (MOEA) known as NSGA-II (Non-dominated Sorting Genetic Algorithm-II) has also been

modified in this paper. The modification includes the introduction of a mutation algorithm which has been applied over the entire population of chromosomes. The formulated problem has been solved by both the original NSGA-II and the modified NSGA-II. The results from both the algorithms have been compared and it has been observed that the modified NSGA-II shows comparatively better results than the original NSGA-II.

REFERENCES

- [1] W. J. Xia and Z. M. Wu, "An Effective Hybrid Optimization Approach for Multi-Objective Flexible Job-Shop Scheduling Problems," *Computers & Industrial Engineering*, Vol. 48, No. 2, 2005, pp. 409-425. [doi:10.1016/j.cie.2005.01.018](https://doi.org/10.1016/j.cie.2005.01.018)
- [2] L.-N. Xing, Y.-W. Chen and K.-W. Yang, "Multi-Objective Flexible Job Shop Schedule: Design and Evaluation by Simulation Modeling," *Applied Soft Computing*, Vol. 9, No. 1, 2009, pp. 362-376. [doi:10.1016/j.asoc.2008.04.013](https://doi.org/10.1016/j.asoc.2008.04.013)
- [3] C. X. Miao, Y. Z. Zhang and Z. G. Cao, "Bounded Parallel-Batch Scheduling on Single and Multi Machines for Deteriorating Jobs," *Information Processing Letters*, Vol. 111, No. 16, 2011, pp. 798-803. [doi:10.1016/j.ipl.2011.05.018](https://doi.org/10.1016/j.ipl.2011.05.018)
- [4] J. Q. Li, Q. K. Pan and S. X. Xie, "An Effective Shuffled Frog-Leaping Algorithm for Multi-Objective Flexible Job Shop Scheduling Problems," *Applied Mathematics and Computation*, Vol. 218, No. 18, 2012, pp. 9353-9371. [doi:10.1016/j.amc.2012.03.018](https://doi.org/10.1016/j.amc.2012.03.018)
- [5] G. H. Zhang, X. Y. Shao, P. G. Li and L. Gao, "An Effective Hybrid Particle Swarm Optimization Algorithm for Multi-Objective Flexible Job-Shop Scheduling Problem," *Computers & Industrial Engineering*, Vol. 56, No. 4, 2009, pp. 1309-1318. [doi:10.1016/j.cie.2008.07.021](https://doi.org/10.1016/j.cie.2008.07.021)
- [6] J.-Q. Li, Q.-K. Pan and Y.-C. Liang, "An Effective Hybrid Tabu Search Algorithm for Multi-Objective Flexible Job-Shop Scheduling Problems," *Computers & Industrial Engineering*, Vol. 59, No. 4, 2010, pp. 647-662. [doi:10.1016/j.cie.2010.07.014](https://doi.org/10.1016/j.cie.2010.07.014)
- [7] J. Gao, L. Y. Sun and M. Gen, "A Hybrid Genetic and Variable Neighborhood Descent Algorithm for Flexible Job Shop Scheduling Problems," *Computers & Operations Research*, Vol. 35, No. 9, 2008, pp. 2892-2907. [doi:10.1016/j.cor.2007.01.001](https://doi.org/10.1016/j.cor.2007.01.001)
- [8] A. Rahimi-Vahed and A. H. Mirzaei, "A Hybrid Multi-Objective Shuffled Frog-Leaping Algorithm for a Mixed-Model Assembly Line Sequencing Problem," *Computers & Industrial Engineering*, Vol. 53, No. 4, 2007, pp. 642-666. [doi:10.1016/j.cie.2007.06.007](https://doi.org/10.1016/j.cie.2007.06.007)
- [9] A. R. Rahimi-Vahed, M. Rabbani, R. Tavakkoli-Moghaddam, S. A. Torabi and F. Jolai, "A Multi-Objective Scatter Search for a Mixed-Model Assembly Line Sequencing Problem," *Advanced Engineering Informatics*, Vol. 21, No. 1, 2007, pp. 85-99. [doi:10.1016/j.aei.2006.09.007](https://doi.org/10.1016/j.aei.2006.09.007)
- [10] J. C. Tay and N. B. Ho, "Evolving Dispatching Rules Using Genetic Programming for Solving Multi-Objective Flexible Job-Shop Problems," *Computers & Industrial Engineering*, Vol. 54, No. 3, 2008, pp. 453-473. [doi:10.1016/j.cie.2007.08.008](https://doi.org/10.1016/j.cie.2007.08.008)
- [11] D. Y. Sha and H.-H. Lin, "A Multi-Objective PSO for Job-Shop Scheduling Problems," *Expert Systems with Applications*, Vol. 37, No. 2, 2010, pp. 1065-1070. [doi:10.1016/j.eswa.2009.06.041](https://doi.org/10.1016/j.eswa.2009.06.041)
- [12] D. C. Mattfeld and C. Bierwirth, "An Efficient Genetic Algorithm for Job Shop Scheduling with Tardiness Objectives," *European Journal of Operational Research*, Vol. 155, No. 3, 2004, pp. 616-630. [doi:10.1016/S0377-2217\(03\)00016-X](https://doi.org/10.1016/S0377-2217(03)00016-X)
- [13] T. K. Varadharajan and C. Rajendran, "A Multi-Objective Simulated-Annealing Algorithm for Scheduling in Flowshops to Minimize the Makespan and Total Flowtime of Jobs," *European Journal of Operational Research*, Vol. 167, No. 3, 2005, pp. 772-795. [doi:10.1016/j.ejor.2004.07.020](https://doi.org/10.1016/j.ejor.2004.07.020)
- [14] B. Yagmahan and M. M. Yenisey, "A Multi-Objective ant Colony System Algorithm for Flow Shop Scheduling Problem," *Expert Systems with Applications*, Vol. 37, No. 2, 2010, pp. 1361-1368. [doi:10.1016/j.eswa.2009.06.105](https://doi.org/10.1016/j.eswa.2009.06.105)
- [15] J. Behnamian, S. M. T. Fatemi Ghomi and M. Zandieh, "A Multi-Phase Covering Pareto-Optimal Front Method to Multi-Objective Scheduling in a Realistic Hybrid Flowshop Using a Hybrid Metaheuristic," *Expert Systems with Applications*, Vol. 36, No. 8, 2009, pp. 11057-11069. [doi:10.1016/j.eswa.2009.02.080](https://doi.org/10.1016/j.eswa.2009.02.080)
- [16] T.-C. Chiang, H.-C. Cheng and L.-C. Fu, "NNMA: An Effective Memetic Algorithm for Solving Multiobjective Permutation Flow Shop Scheduling Problems," *Expert Systems with Applications*, Vol. 38, No. 5, 2011, pp. 5986-5999. [doi:10.1016/j.eswa.2010.11.022](https://doi.org/10.1016/j.eswa.2010.11.022)
- [17] Z. Lian, "A United Search Particle Swarm Optimization Algorithm for Multiobjective Scheduling Problem," *Applied Mathematical Modelling*, Vol. 34, No. 11, 2010, pp. 3518-3526. [doi:10.1016/j.apm.2010.03.001](https://doi.org/10.1016/j.apm.2010.03.001)
- [18] R. Tavakkoli-Moghaddam, A. Rahimi-Vahed and A. H. Mirzaei, "A Hybrid Multi-Objective Immune Algorithm for a Flow Shop Scheduling Problem with Bi-Objectives: Weighted Mean Completion Time and Weighted Mean Tardiness," *Information Sciences*, Vol. 177, No. 22, 2007, pp. 5072-5090. [doi:10.1016/j.ins.2007.06.001](https://doi.org/10.1016/j.ins.2007.06.001)
- [19] M. M. Mazdeh, F. Zaerpour, A. Zareei and A. Hajinezhad, "Parallel Machines Scheduling to Minimize Job Tardiness and Machine Deteriorating Cost with Deteriorating Jobs," *Applied Mathematical Modelling*, Vol. 34, No. 6, 2010, pp. 1498-1510. [doi:10.1016/j.apm.2009.08.023](https://doi.org/10.1016/j.apm.2009.08.023)
- [20] H. F. Lewis and S. A. Slotnick, "Multi-Period Job Selection: Planning Workloads to Maximize Profit," *Computers & Operations Research*, Vol. 29, No. 8, 2002, pp. 1081-1098. [doi:10.1016/S0305-0548\(00\)00105-2](https://doi.org/10.1016/S0305-0548(00)00105-2)
- [21] K. R. Baker and B. Keller, "Solving the Single-Machine Sequencing Problem Using Integer Programming," *Computers & Industrial Engineering*, Vol. 59, No. 4, 2010, pp. 730-735. [doi:10.1016/j.cie.2010.07.028](https://doi.org/10.1016/j.cie.2010.07.028)
- [22] L. Lin and H. Jia-Zhen, "Multi-Objective Flexible Job-Shop Scheduling Problem in Steel Tubes Production,"

- Systems Engineering—Theory & Practice*, Vol. 29, No. 8, 2009, pp. 117-126.
- [23] B. Yagmahan and M. M. Yenisey, "Ant Colony Optimization for Multi-Objective Flow Shop Scheduling Problem," *Computers & Industrial Engineering*, Vol. 54, No. 3, 2008, pp. 411-420. [doi:10.1016/j.cie.2007.08.003](https://doi.org/10.1016/j.cie.2007.08.003)
- [24] R.-H. Huang, "Multi-Objective Job-Shop Scheduling with Lot-Splitting Production," *International Journal of Production Economics*, Vol. 124, No. 1, 2010, pp. 206-213. [doi:10.1016/j.ijpe.2009.10.026](https://doi.org/10.1016/j.ijpe.2009.10.026)
- [25] R. Zhang and C. Wu, "A Hybrid Immune Simulated Annealing Algorithm for the Job Shop Scheduling Problem," *Applied Soft Computing*, Vol. 10, No. 1, 2010, pp. 79-89. [doi:10.1016/j.asoc.2009.06.008](https://doi.org/10.1016/j.asoc.2009.06.008)
- [26] L. Wang, Q.-K. Pan, P. N. Suganthan, W.-H. Wang and Y.-M. Wang, "A Novel Hybrid Discrete Differential Evolution Algorithm for Blocking Flow Shop Scheduling Problems," *Computers & Operations Research*, Vol. 37, No. 3, 2010, pp. 509-520. [doi:10.1016/j.cor.2008.12.004](https://doi.org/10.1016/j.cor.2008.12.004)
- [27] E. Moradi, S. M. T. Fatemi Ghomi and M. Zandieh, "Bi-Objective Optimization Research on Integrated Fixed Time Interval Preventive Maintenance and Production for Scheduling Flexible Job-Shop Problem," *Expert Systems with Applications*, Vol. 38, No. 6, 2011, pp. 7169-7178. [doi:10.1016/j.eswa.2010.12.043](https://doi.org/10.1016/j.eswa.2010.12.043)
- [28] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computing*, Vol. 6, No. 2, 2002, pp. 182-197. [doi:10.1109/4235.996017](https://doi.org/10.1109/4235.996017)
- [29] N. Lakshminarasimman, S. Baskar, A. Alphones and M. W. Iruthayarajan, "Multiobjective Mobile Antenna Location Identification Using Evolutionary Optimization Algorithm," *Second International conference on Computing, Communication and Networking Technologies*, Chettinad, 29-31 July 2010, pp. 1-4. [doi:10.1109/ICCCNT.2010.5591640](https://doi.org/10.1109/ICCCNT.2010.5591640)
- [30] H. Ishibuchi, N. Tsukamoto and Y. Nojima, "Empirical Analysis of Using Weighted Sum Fitness Functions in NSGA-II for Many-Objective 0/1 Knapsack Problems," *UKSim 2009: 11th International Conference on Computer Modelling and Simulation*, Cambridge, 25-27 March 2009, pp. 71-76.
- [31] A. Kumar, D. Sharma and K. Deb, "A Hybrid Multi-Objective Optimization Procedure Using PCX Based NSGA-II and Sequential Quadratic Programming," *IEEE Congress on Evolutionary Computation*, 25-28 September, 2007, Singapore, pp. 3011-3018.
- [32] Y. Liu, C. Zhou and W. J. Ye, "A Fast Optimization Method of Using Nondominated Sorting Genetic Algorithm (NSGA-II) and 1-Nearest Neighbor (1 NN) Classifier for Numerical Model Calibration," *IEEE International Conference on Granular Computing*, Beijing, 25-27 July 2005, pp. 544-549.
- [33] M. C. Wang, G. M. Dai, H. P. Hu and M. C. Wang, "Improved NSGA-II Algorithm for Optimization of Constrained Functions," *International Conference on Machine Vision and Human-machine Interface*, TBD Kaifeng, 24-25 April 2010, pp. 673-675.
- [34] M. T. Jensen, "Reducing the Run-Time Complexity of Multiobjective EAs: The NSGA-II and Other Algorithms," *IEEE Transactions on Evolutionary Computing*, Vol. 7, No. 5, 2003, pp. 503-515. [doi:10.1109/TEVC.2003.817234](https://doi.org/10.1109/TEVC.2003.817234)
- [35] M. Sato, H. E. Aguirre and K. Tanaka, "Effects of δ -Similar Elimination and Controlled Elitism in the NSGA-II Multiobjective Evolutionary Algorithm," *IEEE Congress on Evolutionary Computation*, Vancouver, 16-21 July 2006, pp. 1164-1171.
- [36] L. Yu, P. Wang and H. S. Zhu, "A Novel Diversity Preservation Strategy based on Ranking Integration for Solving Some Specific Multi-Objective Problems," *Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science*, Hong Kong, 10-12 August 2010, pp. 97-101. [doi:10.1109/DCABES.2010.145](https://doi.org/10.1109/DCABES.2010.145)
- [37] M. R. Mansour, A. C. Santos, J. B. London Jr., A. C. B. Delbem and N. G. Bretas, "Node-Depth Encoding and Evolutionary Algorithms Applied to Service Restoration in Distribution Systems," *IEEE Power and Energy Society General Meeting*, Minneapolis, 25-29 July 2010, pp. 1-8.